1. Create an assert statement that throws an AssertionError if the variable spam is a negative integer.

assert spam >= 0

2. Write an assert statement that triggers an AssertionError if the variables eggs and bacon contain strings that are the same as each other, even if their cases are different (that is, 'hello' and 'hello' are considered the same, and 'goodbye' and 'GOODbye' are also considered the same).

assert eggs.lower() != bacon.lower()

3. Create an assert statement that throws an AssertionError every time.

assert False

4. What are the two lines that must be present in your software in order to call logging.debug()?

import logging

logging.basicConfig(filename = "Assignment.log", level=logging.DEBUG)

5. What are the two lines that your program must have in order to have logging.debug() send a logging message to a file named programLog.txt?

import logging

logging.basicConfig(filename = "Log.txt", level=logging.DEBUG)

6. What are the five levels of logging?

The five levels of logging are :

DEBUG

INFO

WARNING

ERROR

CRITICAL

7. What line of code would you add to your software to disable all logging messages?

logging.disable(logging. DEBUG)

8.Why is using logging messages better than using print() to display the same message?

logging messages offers more flexibility, control, and convenience compared to print().

9. What are the differences between the Step Over, Step In, and Step Out buttons in the debugger?

Step In : In this if any function comes in way, during the program running , debugger also look into the code of that function and after fully completing function it moved to function call line , and move to next code of line.

Step Over : Skip code of Function , which comes in way , and moves to next line

Step Out: In this it comes out quickly of function , and return to where it called.

10.After you click Continue, when will the debugger stop ?

The debugger will stop again either at the next breakpoint you set or when your program finishes running.

11. What is the concept of a breakpoint?

Breakpoint is as a temporary stop sign that you place in your code. Whenever program reaches that point, it pauses and enable us to take a closer look at what happening in code. Breakpoints help us to find & fix issues in our code by giving a chance to investigate line by line.